Los Alamos
NATIONAL LABORATORY
research note

Applied Theoretical & Computational Physics
Diagnostic Applications Group (X–5)

To/MS: Distribution
From/MS: R. E. Prael, X–5, MS F663
Phone/FAX: 7–7283/5–3046
Email: rep@lanl.gov
Symbol: X–5–RN (U) 00–30
LA-UR-00-4027
Date: August 23, 2000

Proposed Modification to the Charged Hadron Tracking Algorithm in MCNPX

Summary

Unreasonable results observed in the calculation of a depth/dose curve in water using MCNPX have lead to a reexamination of the logic applied to energy loss and energy straggling while tracking a charged hadron through material. A modified version of subroutine CHG_PL has been developed and tested which provides a high degree of consistency with LAHET3 calculations.

I. Introduction

The original problem[1] in which the difficulty was detected was specified as a 157.2 MeV proton beam on water. The protons were tracked through a geometry of planes with 1 mm spacing; energy deposition (F8 tally) was computed in each 1 mm-thick region. To obtain the best possible comparison between LAHET3 and MCNPX, the original input was modified to transport protons with only multiple scattering, slowing down, and energy straggling, and with no nuclear nonelastic or elastic events. The magnitude of the discrepancy is readily discernible in Figure 1.

It is worth noting what *does not* contribute to the problem. First, LAHET3 and MCNPX use the same coding for multiple scattering, applied at steps that are at least comparable. Second, the mean stopping power used by MCNPX within an energy interval of the range tabulation ("energy step") is obtained by integration over the interval; apart from numerics, the method is equivalent to linear interpolation in a range table in LAHET3 application. The pointwise evaluation of the stopping power is the same in both codes.

For purposes of methods testing and validation, proton surface flux tallies at 5 cm, 10 cm and 15 cm depth were also made. The energy distribution of the surface flux is particularly sensitive to application of the energy straggling algorithm. Since the flux on the specified surfaces and the location and shape of the Bragg peak should be independent of any specified intervening geometry, an input with a second geometry was defined, with three regions each 5 cm thick, followed by the same 1 mm-thick regions near the end of range. In the original problem, a proton would be tracked across 150 surfaces to reach a depth of 15 cm, while in the second case it would cross only three.

II. Tracking Charged Particles in MCNPX

The logic of tracking a charged particle in subroutine CHG_PL of MCNPX may be described by the following (over)simplified sequence.

1. A charged particle with energy E is assigned to energy step i, where $E_i > E \geq E_{i+1}$.
2. A random energy straggling increment $\Delta(E, p)$ is sampled for a step size p defined by

$$p_i = \frac{E_i - E_{i+1}}{q_i}$$

   where q_i is the mean stopping power in energy interval i.
3. An effective stopping power $\tilde{q}_i$ is defined for the current interval by

$$\tilde{q}_i = q_i + \Delta(E, p)/p$$

   and the "substep distance" is defined as $\delta = p/n$ where n is the number of substeps per energy interval (ESTEP, default =3).
4. A loop is initiated over the n substeps.
5. The distance to interaction d_I, distance to time cutoff d_t, distance to energy cutoff d_c and distance to surface crossing d_s are obtained.
6. The tracking distance d is defined by

$$d = \min(\delta, d_s, d_I, d_t, d_c)$$

   and the particle is advanced by d.
7. The multiple scattering is applied to the directional coordinates for E and d.
8. The new energy is defined by $E - \tilde{q}_i d$.
9. If $d < \delta$ or $E < E_{i+1}$ (and the particle remains to be tracked), the energy interval index i is incremented and the procedure is reinitiated at step (1) above; otherwise, the substep loop is continued.

Approximations which may cause difficulties are introduced at (3) and at (8) above.

The use of $\tilde{q}_i$ gives a correct representation of energy straggling *only* when the loop over all n substeps is exhausted. Surface crossings, interactions, and exiting the energy interval in less than n complete steps will all introduce some error. Tracking through complex geometry will exaggerate the cumulative error. The method is completely inappropriate for the estimation of energy spreading when passing through a single optically thin region such as a foil.

Adjusting the energy at (8) above introduces an error at every application in which the energy drops out of the current energy interval; in this case, part of the tracking distance d is actually in an energy range where the stopping power may be greatly different from the $\tilde{q}_i$ applied. The effect of the approximation is mitigated by many surface crossings, by frequent scattering events, or by increasing the number of substeps n. However, at low energies where the stopping power increases with decreasing energy, the energy loss over an extended range will be consistently underestimated; at high energies where the stopping power decreases with decreasing energy, overestimation of energy loss occurs.

III. The Modified Tracking Algorithm

The modified algorithm for calculating the mean value new energy E' after a tracking length d starting from initial energy E in energy interval i is given by the sequence of equations:

$$E' = E - q_i d \quad \text{for} \quad E' > E_{i+1}$$

$$E' = E_{i+1} - q_{i+1}\left(d - \frac{E - E_{i+1}}{q_i}\right) \quad \text{for} \quad E_{i+1} \geq E' > E_{i+2}$$

and for $E_{i+m} \geq E' > E_{i+m+1}$with $m > 1$,

$$d' = d - \frac{E - E_{i+1}}{q_i} - \frac{E_{i+1} - E_{i+2}}{q_{i+1}} - \ldots - \frac{E_{i+m-1} - E_{i+m}}{q_{i+m-1}}$$

$$E = E_{i+m} - q_{i+m} d'$$

The energy straggling correction is obtained by applying the sampling for the energy correction whenever the energy is recalculated, using the real tracking length d and the initial energy E. Thus, if E' is obtained as above, the new energy at the end of the step is

$$E'' = E' - \Delta(E, d)$$

The two corrections may be applied separately, as shown in the examples.

IV. Results

As previously noted, Figure 1 shows comparative results for the calculation as originally presented. Figure 2 provides detail of the location and shape of the energy deposition peak. Note that applying the energy correction alone makes little difference, since the energy is adjusted at the many surface crossings in the geometry defined by planes at 1 mm spacing; the straggling correction is essential, however, for correct execution of the calculation so specified. When the complete modification is applied, the computed results are essentially identical whether a track crosses 150 or 3 surfaces in the first 15 cm of travel.

Looking at the flux distribution on a plane is a more sensitive test of the effect of these modifications. In Figure 3, the proton flux is shown on the plane 15 cm deep into the water after 150 surface crossings. Again, due to the many surface crossings, the energy correction add little; the modification agrees very well with the LAHET tracking algorithm. A completely different picture appears in Figure 4, where the flux at 15 cm depth is obtained after only 3 surface crossing. In this case, it is the energy correction that is essential. The straggling correction adds little, since the current method applies straggling correctly when the distance of an energy step is much smaller than the thickness of a region (as defined by the explicit geometry). *Note:* in all the plots where "flux" is shown, it actually has the units $10(\text{cm})^{-2}(\text{MeV})^{-1}$.

Additional examples are shown in Figures 5 and 6. In the former, the flux is shown at 1 cm depth, shorter than a full energy step. The asymmetric form of the flux distribution is noticeable. In Figure 6, the flux is shown for a very thin region 1 mm thick. The shape is characteristic of the Landau distribution. The agreement between the LAHET3 results and the results from modified MCNPX appear to validate the implementation of the energy straggling models. The limitations of the numerical methods in this case are also obvious.

As a final part of this evaluation, the results shown in Figure 7 are presented. In this case, the sensitivity of the flux calculation to the number of substeps per energy step is presented. The results are somewhat disconcerting. Further review of the new coding may reveal some correctable limitation sensitive to step size. Little of the fluctuation in shape is statistical; the *estep = 30* case was run for 6.5x10^5 histories, the other cases for 10^6. The *shift* in the flux peak is more significant. It is likely to result from the know fact that the sampling scheme is not unbiased. From numerical limitations, the mean value $\|\Delta(E,p)\| \neq 0$, and the bias varies considerably over the range of parameters included in this comparison. Further documentation of the numerical properties of the straggling routines is called for. There are six different algorithms covering the full range of usage. The current tests reached four of them, although the cases for $estep = 1, 3, 6$ employed the same one over most samplings.

V. Conclusions and Recommendations

The modified algorithm presented here has been tested only on the specific problem described above. Further testing should be conducted for a variety of geometries, compositions and energies to insure proper implementation. However, the current method provides a correct application on each track length segment, including substeps, within the limitations of the models for energy loss, energy straggling and multiple scattering. It is in principal independent of geometry and is effective for thin as well as thick regions. It should be fully compatible with tallying in an overlaid mesh geometry, although that has not been tested here. As such, it should be used as the “base case” from which further approximations, simplifications and efficiencies may be applied.

For the conditions of the test case (no nuclear interactions), and using the default 3 substeps per energy step, the full modification increased the execution time by 67% for the thick-cell (5 cm) geometry and by 53% for the thin-cell (1 mm) geometry. For the straggling correction alone, the increase was 45% and 40% respectively. Applying the energy correction alone actually *reduced* execution time, 15% and 3% respectively, because the correct energy loss leads to earlier termination.

A more efficient algorithm may be constructed by applying the energy straggling with an accumulated patch length after, for instance, a full energy step (and at each surface crossing or interaction, of course), rather than each substep. Theoretically it is a valid application, but should be made an input option since it could have adverse effects on mesh tallying; all the variance in the energy straggling would be applied on the *last* substep!

Efficiency could also be improved by reducing the need for a large number of substeps per step, the motivation for which is both the desire to minimize the energy change between applications of multiple scattering and to minimize the transverse tracking error on each path segment. It may be possible to define an "effective energy" over a path segment that will allow an accurate application of multiple scattering over larger step sizes. Furthermore, the use of a "transverse longitudinal correction (TLC)" in tracking might minimize the need for many substeps.

To conclude, here are a few minor suggestions to be considered.

- Attempting to execute the energy straggling routines in the large step ("Gaussian") regime should result in at least a warning error; if necessary, a limiting step length may be simply applied to prevent such an occurrence.
- Allow separate substep factors for each call of particles by mass: electrons, mesons and baryon (perhaps *estep, mstep* and *pstep*).
- Remove the warning message for *estep* < 3, at least when electrons are not being transported, but keep the default at 3.
- Extend this comparison to examine the estimation of the angular dispersion as a function of depth in target.
- Use DBCN control of *both* multiple scattering *and* energy straggling for the foreseeable future to facilitate testing.

REFERENCES

[1] J. V. Siebers, Medical College of Virginia, Virginia Commonwealth University (personal communication).

[2] H. G. Hughes, R. E. Prael, and R. C. Little, "MCNPX – The LAHET/MCNP Code Merger", X-Division Research Note XTM-RN(U)97-012, LA-UR-97-4891, Los Alamos National Laboratory (April 1997).

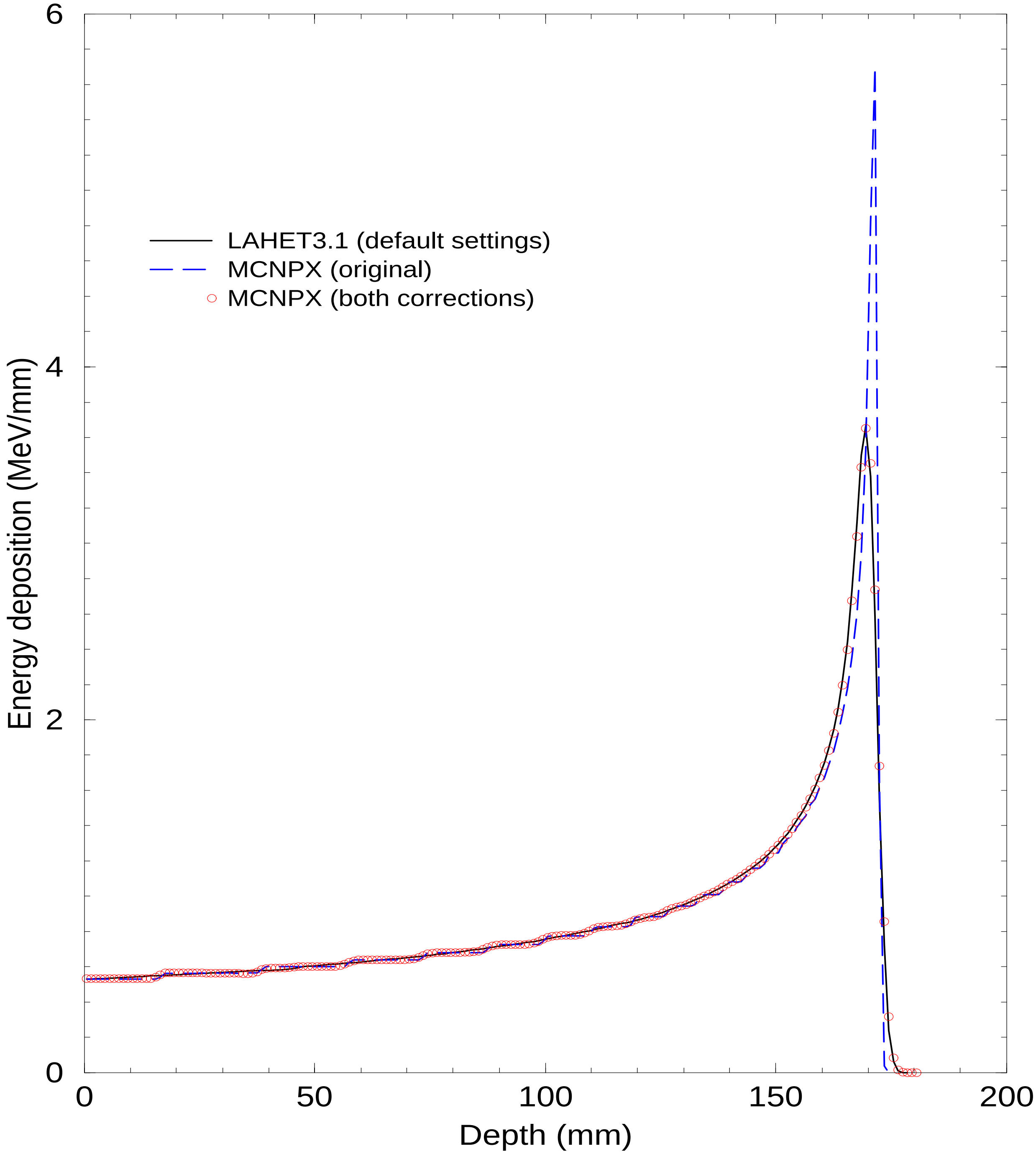

Figure 1: Energy deposition from 157.2 MeV proton beam in water, tracked through 1 mm thick regions.

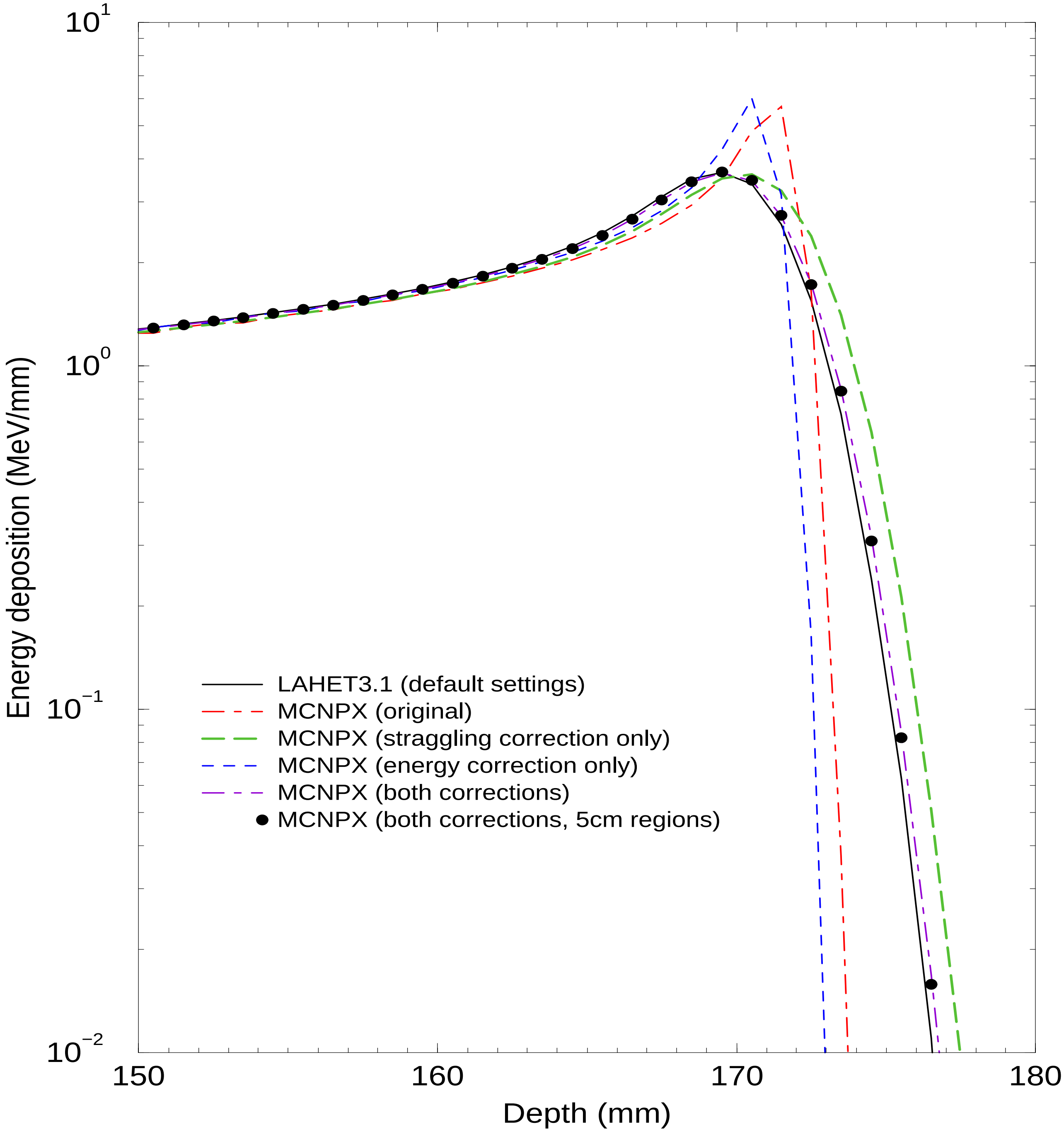

Figure 2: Energy deposition from 157.2 MeV proton beam in water, tracked through 1 mm thick regions except as indicated. The specific effect of each modification is shown.

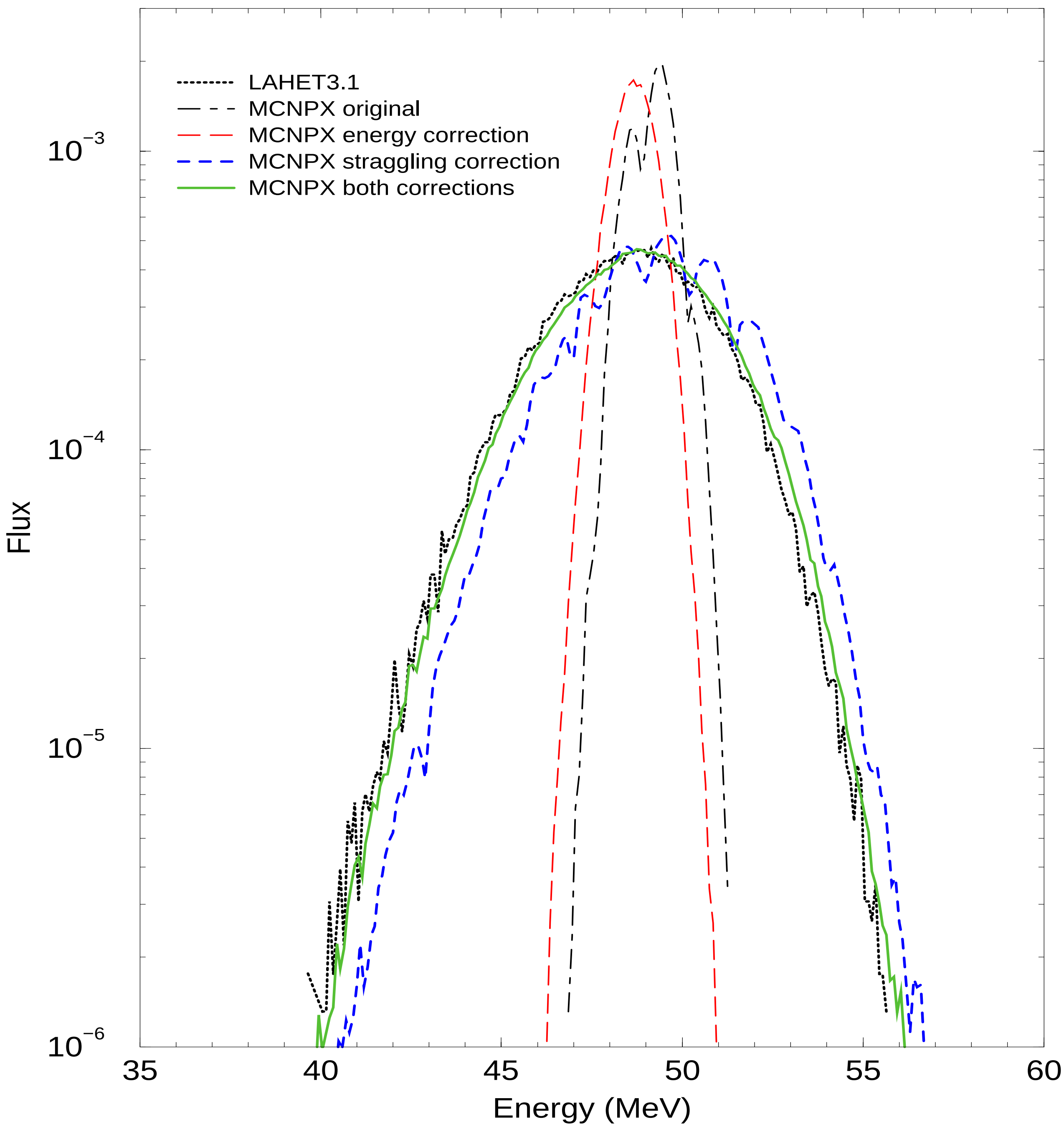

Figure 3: Flux distribution from 157.2 MeV proton beam at 15 cm depth in water, tracked through *1 mm* thick regions using estep=3. The specific effect of each modification is shown.

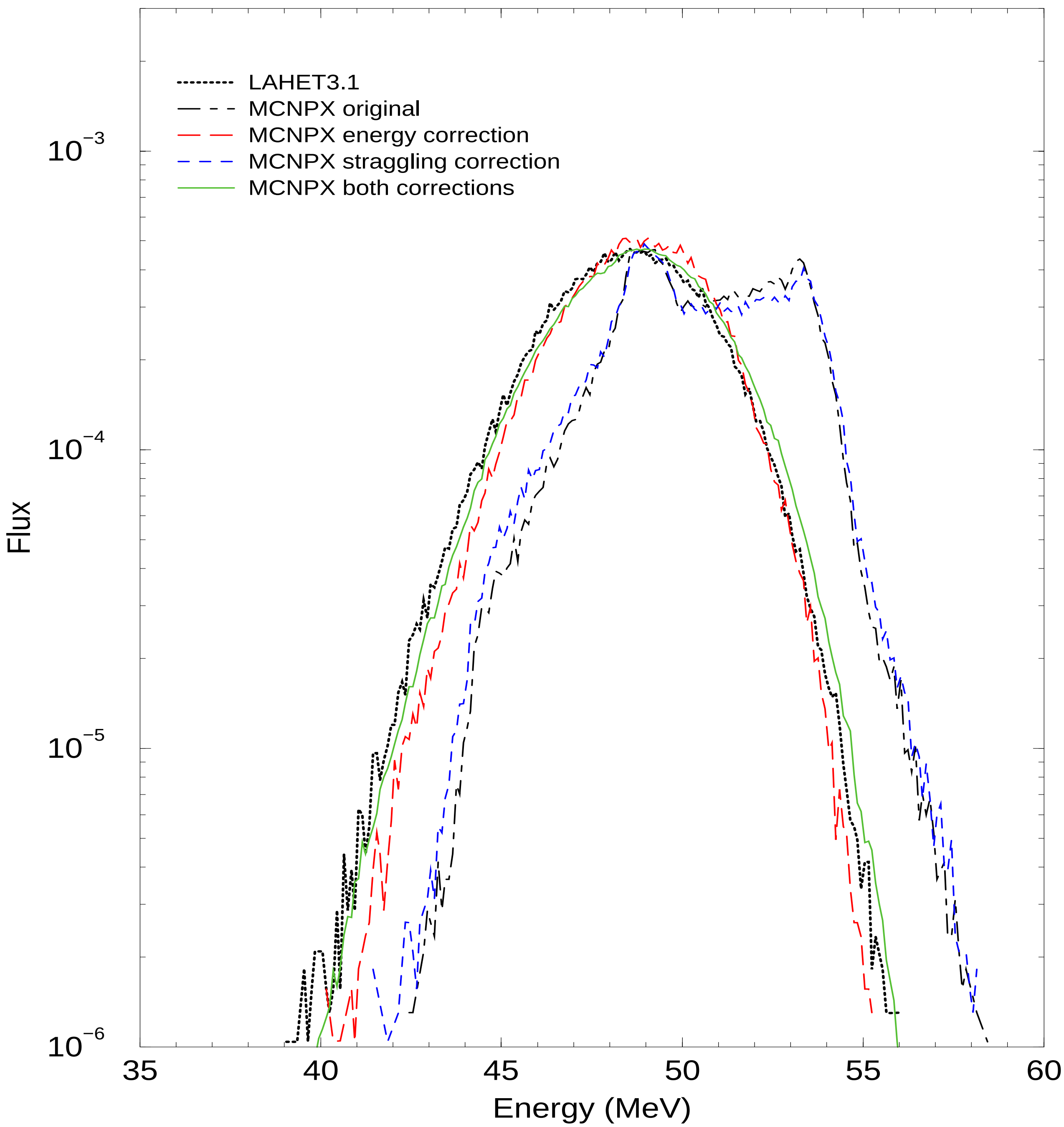

Figure 4: Flux distribution from 157.2 MeV proton beam at 15 cm depth in water, tracked through *5 cm* thick regions using estep=3. The specific effect of each modification is shown.

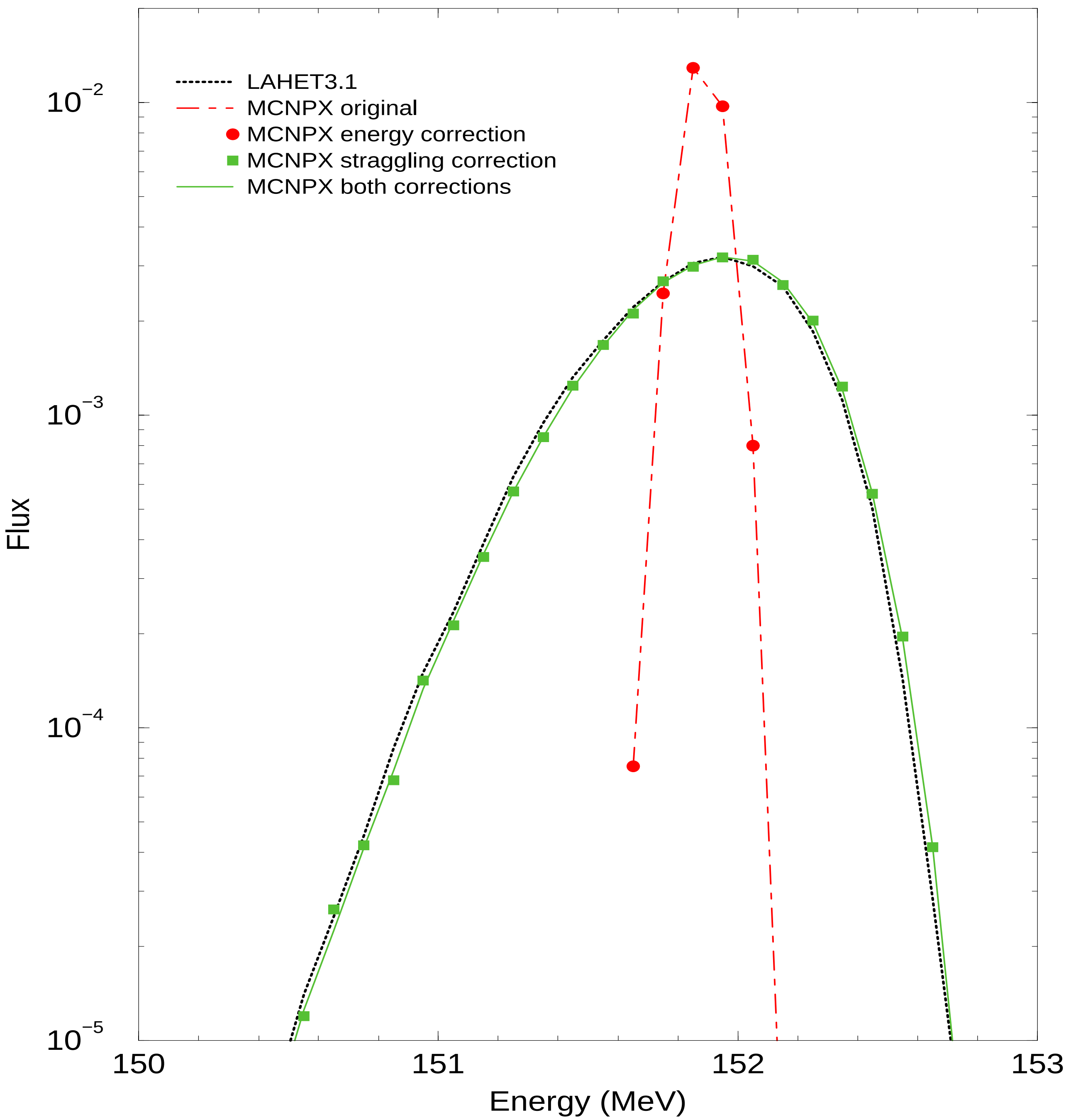

Figure 5: Flux distribution from 157.2 MeV proton beam at 1 cm depth in water, tracked through *1 mm* thick regions using estep=3. The specific effect of each modification is shown.

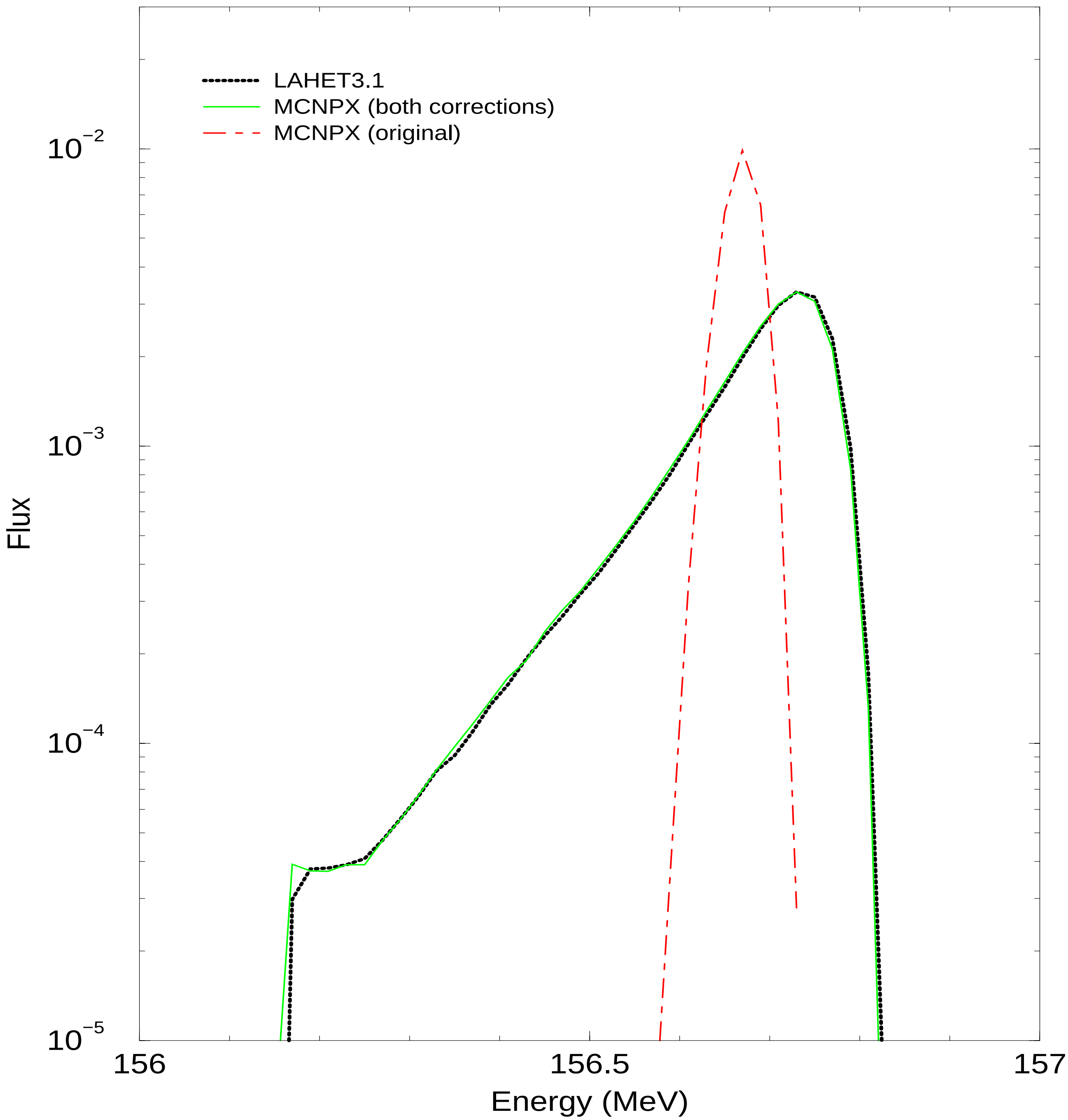

Figure 6: Flux distribution from 157.2 MeV proton beam at 1 mm depth. The specific effect of each modification is shown.

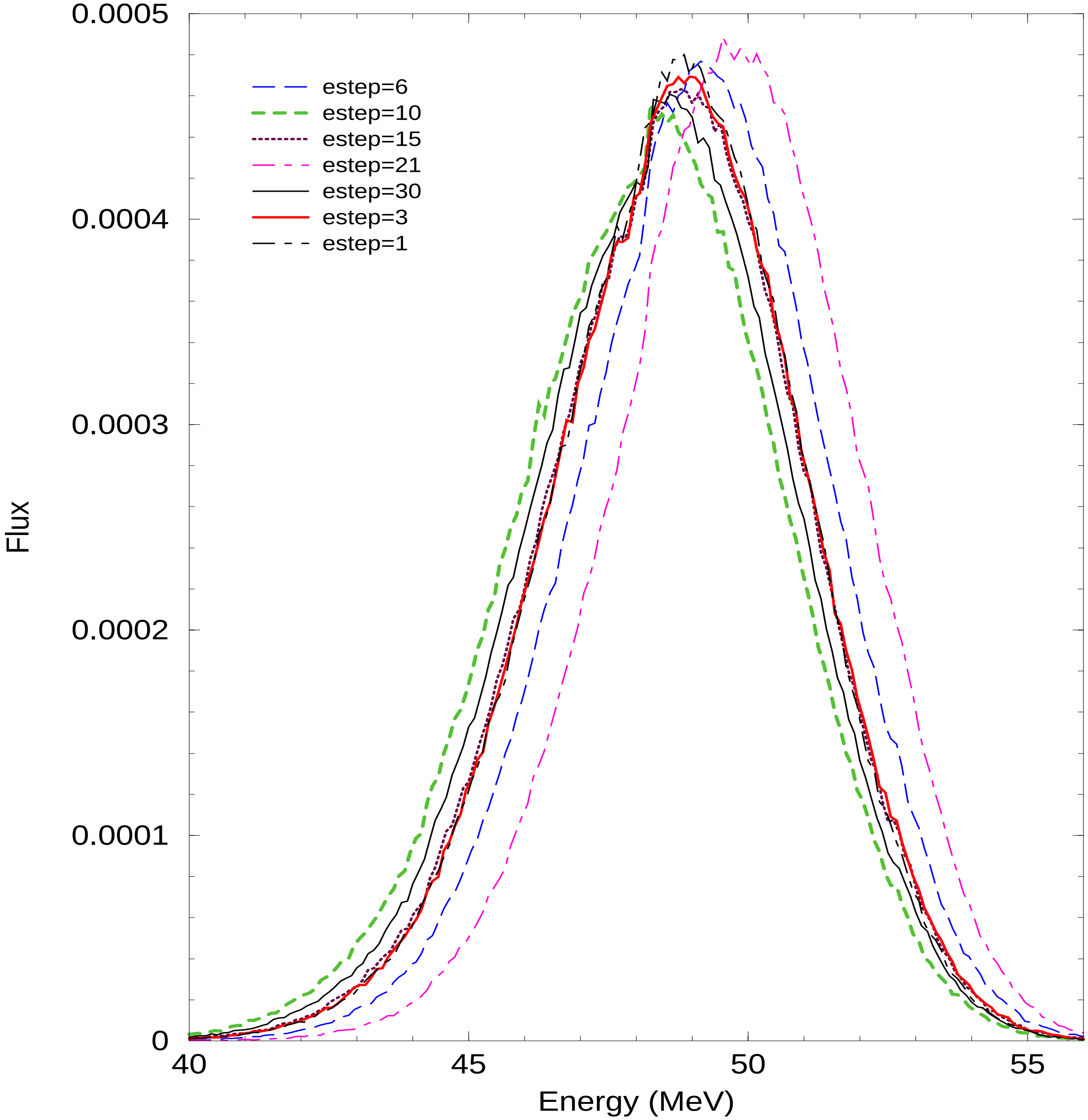

Figure 7: Effect of *ESTEP* value on flux at 15 cm depth in water, tracked through *5 cm* thick regions.

Listing of modifications to subroutine chg_pl.F (enclosed by CREP comments)

```
c_deck chg_pl chg_pl                                                    chg_pl     1
      subroutine chg_pl                                                 chg_pl     2
             *
             *
             *
c                                                                       chg_pl    43
c        begin an energy step (a set of substeps).                      chg_pl    44
   40 pmf=huge                                                          chg_pl    45
c        Initializing qs is probably unnecessary.
      qs=0.0
      if(mkc.ne.0) then                                                 chg_pl    46
         pmf=drs(ldrs+ipt,ngp+nee_max*(mkc-1))/rho(lrho+icl)            chg_pl    47
CREP begin #1
         qs=qav(lqav+ipt,ngp+nee_max*(mkc-1))*rho(lrho+icl)
C        p=pmf*nsb(lnsb+mkc)                                            chg_pl    48
C        DEBUG: counting of resamplings of straggling.
C        iz = 0
C  50    call glando(rang,p,avz(lavz+mkc),ava(lava+mkc),den(lden+icl),  chg_pl    49
C    1               erg,gpt(ipt),de,ii)                                chg_pl    50
C        qs=qav(lqav+ipt,ngp+nee_max*(mkc-1))*rho(lrho+icl)+de/p        chg_pl    51
C        if(qs.le.0.) then
C           iz = iz + 1
C           if(iz.gt.1000) then
C              call expirx(1,'chg_pl',
C    1            'more than 1000 resamplings of glando.')
C              return
C           endif
C           go to 50
C        endif
CREP end #1
      endif                                                             chg_pl    52
c                                                                       chg_pl    53
c        substep loop.                                                  chg_pl    54
      do 150 ns=nm,1,-1                                                 chg_pl    55
      if(wgt.le.0.) then                                                chg_pl    56
         call expirx(1,'chg_pl',                                        chg_pl    57
     1  'the weight of the current particle is zero or less.')          chg_pl    58
         return                                                         chg_pl    59
      endif                                                             chg_pl    60
      nch(ipt)=nch(ipt)+1                                               chg_pl    61
      ncp=ncp+1                                                         chg_pl    62
      n1=ngp                                                            chg_pl    63
#ifdef MESHTAL
      f_ed=0.0
#endif
c                                                                       chg_pl    64
c        calculate distances to time and energy cutoff.                 chg_pl    65
      dtc=vel*(tco(ipt)-tme)                                            chg_pl    66
      dc=huge                                                           chg_pl    67
      if(mkc.ne.0)dc=(erg-elc(ipt))/qs                                  chg_pl    68
c                                                                       chg_pl    69
c        calculate the distance to interaction.                         chg_pl    70
```

```
      di=huge                                                           chg_pl
      if(ipt.eq.9.and.jan.eq.1) then                                    chg_pl
         call ace_tot(erg,lp,mkc,nel(lnel+1),mel(lmel+1),aa9(laa9+1),   chg_pl
     1    sigg(lsgg+1,1),hsigg(lhsg+1,1),xsiso(lxso+1,1),esxmp(lesp+1), chg_pl
     2    esxln(lexn+1,1),rho(lrho+icl),di)                             chg_pl
      else                                                              chg_pl
         call get_tot(erg,lp,mkc,nel(lnel+1),mel(lmel+1),aa9(laa9+1),   chg_pl
     1    sigg(lsgg+1,1),hsigg(lhsg+1,1),xsiso(lxso+1,1),rho(lrho+icl), chg_pl
     2    di)                                                           chg_pl
      endif                                                             chg_pl
c                                                                       chg_pl
c        calculate the distance to the cell boundary, dls.              chg_pl
      d1=min(pmf,dtc,di,dc)                                             chg_pl
      if(mbd(lmbd+icl).ne.0.or.jsu.ne.0)go to 60                        chg_pl
      dls=huge                                                          chg_pl
      rr=rr-d1                                                          chg_pl
      if(rr.gt.0.)go to 70                                              chg_pl
      rr=dbmin()-d1                                                     chg_pl
      if(rr.gt.0.)go to 70                                              chg_pl
   60 if(lca(llca+icl).lt.0)call chkcel(icl,3,j)                        chg_pl
      call track(icl)                                                   chg_pl
      if(kdb.ne.0)return                                                chg_pl
      jsu=jap                                                           chg_pl
   70 continue                                                          chg_pl
c                                                                       chg_pl
c        tally the track length in the cell.                            chg_pl
      d=min(dls,d1)                                                     chg_pl
             *
             *
             *
c        scatter the particle.                                          chg_pl
      uold(1)=uuu                                                       chg_pl
      uold(2)=vvv                                                       chg_pl
      uold(3)=www                                                       chg_pl
      if(mkc.ne.0) then                                                 chg_pl
c                                                                       chg_pl
c        TEMPORARY measure: default gaussian; dbcn(21) for no scatter.
         if(dbcn(21).eq.0.) then
            call mscat3(lp,d,arg(larg+mkc),erg,th)
            call rotas(cos(th),uold,uuu,lev,irt)
         endif
c                                                                       chg_pl
c           lose energy to ionization along the track.                  chg_pl
         eg0=erg                                                        chg_pl
CREP begin #2
         d_left=d
         n2=ngp
         qs1=qs
   45    continue
         dps=(erg-eee(leee+ipt,n2+1))/qs1
         if (d_left.le.dps) then
           erg=erg-qs1*d_left
         else
           d_left=d_left-dps
           n2=n2+1
```

```
          erg=eee(leee+ipt,n2)
C         qs1=qs1*qav(lqav+ipt,n2+nee_max*(mkc-1))/
C    1       qav(lqav+ipt,n2-1+nee_max*(mkc-1))
          qs1=qav(lqav+ipt,n2+nee_max*(mkc-1))*rho(lrho+icl)
          go to 45
        endif
c       DEBUG: counting of resamplings of straggling.
        iz = 0
   50   call glando(rang,d,avz(lavz+mkc),ava(lava+mkc),den(lden+icl),
     1              eg0,gpt(ipt),de,ii)
        if((eg0-erg+de).le.0.) then
           iz = iz + 1
           if(iz.gt.1000) then
              call expirx(1,'chg_pl',
     1           'more than 1000 resamplings of glando.')
              return
           endif
           go to 50
        endif
        erg=erg-de
        paxtc(6,11,ipt)=paxtc(6,11,ipt)+wgt*(eg0-erg)
#ifdef MESHTAL
        f_ed=eg0-erg
CREP end #2
        l=locct(llct+1,icl)
        if(l.ne.0)call tally(f_ed,1,l,d)
        if(kdb.ne.0)return
        do 76 i=0,lev-1
           l=locct(llct+1,int(udt(7,i)))
   76      if(l.ne.0)call tally(f_ed,1,l,d)
        if (mtlflg.ne.0.or.medflg.ne.0.and.f_ed.ne.0.) then
           do 78 i=1,6
              if(lev.eq.0) then
                 zd(i)=gpblcm(i)
              else
                 zd(i)=udt(i,0)
              endif
   78         continue
           zd(7)=wgt
           zd(8)=erg
           zd(9)=f_ed
           jz(1)=mkc
           jz(5)=2
           call gdtal(2,d,zd,jz)
        endif
#endif
CREP begin #3
        if(erg.le.elc(ipt)) then
CREP end #3
           call uplpos(xxx,uold,lev,d,vel,1)                            chg_pl
c          to avoid a roundoff problem...                               chg_pl
           erg=elc(ipt)                                                 chg_pl
           nb=2                                                         chg_pl
           go to 500                                                    chg_pl
        endif                                                           chg_pl
```

```
c                                                                  chg_pl   133
         if(erg.lt.eee(leee+ipt,ngp+1)) then                       chg_pl   134
  100       ngp=ngp+1                                              chg_pl   135
            if(erg.lt.eee(leee+ipt,ngp+1))go to 100                chg_pl   136
         endif                                                     chg_pl   137
      endif                                                        chg_pl   138
c                                                                  chg_pl   139
c        move particle to surface, substep, interaction, or termination.  chg_pl   140
c        produce secondary particles.                              chg_pl   141
             *
             *
             *
c        complete energy substep for particle remaining in cell icl.   chg_pl   218
  150 if(n1.ne.ngp)go to 40                                        chg_pl   219
      go to 40                                                     chg_pl   220
c                                                                  chg_pl   221
  500 ec=erg                                                       chg_pl   222
             *
             *
             *
      return                                                       chg_pl   241
      end                                                          chg_pl   242
```